CONTROL METHOD FOR STORAGE SYSTEM, STORAGE SYSTEM, AND STORAGE DEVICE

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to a control method for a storage system, a storage system, and a storage device.

2. Description of the Related Art

Disaster recovery in storage systems is receiving attention. Known techniques for performing disaster recovery include techniques for administrating the replication of data stored in a replication source storage area in a replication destination storage area as well (remote copying) (for example, see Japanese Patent Application Laid-open No. 2001-337939, US Patent Application No. 2003/51111, and US Patent No. 6591351 Specification). By this technique, when an information processing device which accesses the replication source storage device fails, the processing that was being performed on the information processing device can be taken over to another information processing device which accesses the replicated data in the replication destination storage device.

However, copying of the data for replication from the replication source storage device to the replication destination storage device is sometimes not completed by the time the aforementioned takeover occurs. In this case, the information processing device for accessing the replication destination storage device is queued until the aforementioned copying is completed, and in some cases, timeout and other problems occur in the information processing device, and a takeover sometimes fails.

An object of the present invention is to provide a control method for a storage system, a storage system, and a storage device capable of performing smooth takeover of processing between information processing devices.

SUMMARY OF THE INVENTION

The control method for a storage system of present invention for overcoming the foregoing drawbacks consists of a first information processing device; a second information processing device which is connected to the first information processing device so as to be capable of communicating with the first information processing device, and which constitutes a cluster with the first information processing device; a first storage device which is connected to the first information processing device so as to be capable of communicating with the first information processing device, and performs writing/reading of data to a first storage area in the first storage device according to a data input/output request transmitted from the first information processing device; and a second storage device which is connected to the second information processing device so as to be capable of communicating with the second information processing device, and performs writing/reading of data to a second storage area in the second storage device according to a data input/output request transmitted from the second information processing device; wherein the first storage device and the second storage device are connected to each other so as to be capable of communicating with each other; the second storage device, during a first processing in which the first storage device transmits to the second storage device a replication of the data written to the first storage area, and the second storage device that received the data writes the data to the second storage area, requests first information from the first storage device indicating which data written in the first storage area has not yet been transmitted to the second storage device and therefore has not been written to the second storage area, when a failover notice is received from the second information processing device; the second storage device notifies the second information processing device that a data input/output request can be received when the first information is received from the first storage device;

and the second storage device refers the first information upon receipt of a data read request transmitted from the second information processing device in which failover has occurred, requests the target data of the data read request to the first storage device if it ~~is~~ concluded that the target data of the data read request is not stored in the second storage area, and transmits to the second information processing device the target data of the data read request transmitted from the first storage device as per the request.

Also, the problems disclosed in the present application, and the method of overcoming these problems will be described in greater detail in the Embodiments and Drawings sections.

The present invention provides a control method for a storage system, a storage system, and a storage device capable of performing smooth takeover of processing between information processing devices.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG 1 is a block diagram depicting the overall structure of the storage system 90 pertaining to the present embodiment;

FIG 2 is a diagram depicting the specific structure of a disk array device described as an example of the storage devices 30 and 40 pertaining to the present embodiment;

FIG 3 is a block diagram depicting an example of the structure of the information processing devices 10 and 20 pertaining to the present embodiment;

FIG 4 is a block diagram depicting an example of the structure of the channel controller 101 pertaining to the present embodiment;

FIG 5 is a diagram outlining the structure of the storage system 90 described as an embodiment of the present invention;

FIG 6 is a diagram depicting an example of the differential administration tables 1 and 2 (400 and 401) pertaining to the present

embodiment;

FIG 7 is a flowchart describing the processing relating to remote copying from the first storage device 30 to the second storage device 40 pertaining to the present embodiment;

FIG 8 is a diagram depicting an example of the processing performed when the second information processing device 20 detects failure of the first information processing device 10, pertaining to the present embodiment;

FIG 9 is a flowchart depicting an example of the processing performed when the second storage device 40 receives a data write request from the second information processing device 20 after failover in accordance with the present embodiment;

FIG 10 is a flowchart depicting an example of processing performed when the second storage device 40 receives a data read request from the second information processing device 20 after failover in accordance with the present embodiment;

FIG 11 is a diagram outlining the structure of the storage system 90 described as an example of another embodiment;

FIG 12 is a diagram depicting an example of the local unreflected tables 501 and 601 pertaining to another embodiment;

FIG 13 is a diagram depicting an example of the remote unreflected tables 502 and 602 pertaining to another embodiment;

FIG 14 is a diagram depicting an example of the data location determination tables 500 and 600 pertaining to another embodiment;

FIG 15 is a flowchart depicting the remote copying performed in the background, described as an example of another embodiment; and

FIG 16 is a flowchart depicting an example of processing performed in a case in which the second storage device 40 receives a data write request from the second information processing device 20 after failover or failback in accordance with another embodiment.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

FIG. 1 is a block diagram depicting the overall structure of the storage system 90 described as an embodiment of the present invention.

The storage system 90 is made up of a first information processing device 10, second information processing device 20 connected so as to be capable of communicating therewith, first storage device 30 connected so as to be capable of communicating with the first information processing device 10, and a second storage device 40 connected so as to be capable of communicating with the second information processing device 20. The storage system 90 consists, for example, of an online system for a bank, accounting, or other business, an inventory management system in a commercial firm, distributing company, or the like, or a system used regularly for seating reservations in a railroad or airline company. This storage system 90 is constructed for performing disaster recovery in the event of an earthquake, fire, typhoon, lightning strike, terrorist attack, or the like.

The first storage device 30 is connected so as to be capable of communicating with the second storage device 40 via a first network 80. The first network 80 consists, for example, of a Gigabit Internet (registered trademark), ATM (AsynchroNOus Transfer Mode), public circuit, or the like.

The information processing devices 10 and 20 are connected so as to be capable of communicating with the storage devices 30 and 40 respectively via second networks 50 and 60. The each second networks 50 and 60 consists, for example, of a SAN (Storage Area Network). The each SAN consists of a network for exchanging data between the information processing device 10 (or 20) and the storage device 30 (or 40) in the form of "block" units. The storage devices 30 and 40 are block devices and store data in the form of block units. Also, the each second networks 50 and 60 may consist of a LAN (Local Area Network), iSCSI (Internet Small Computer System Interface), Fibre Channel, ESCON (Enterprise Systems Connection) (registered trademark),

FICON (Fiber Connection) (registered trademark), or the like.

The information processing devices 10 and 20 are connected so as to be capable of communicating via a third network 70. The third network 70 consists, for example, of a LAN (Local Area Network).

FIG. 2 depicts the specific structure of a disk array device described as an example of the storage devices 30 and 40. Also, the storage devices 30 and 40 may consist, for example, of semiconductor storage devices or the like, instead of a disk array devices.

The disk array device comprises a storage control device 100 and disk drives 108. The storage control device 100 comprises a channel controller 101, remote communication interface 102, disk controllers 103, shared memory 104, cache memory 105, and a switching controller 106 or the like made up of a crossbar switch connected so as to be capable of communicating with the aforementioned components.

The cache memory 105 is mainly used for temporarily storing data exchanged between the channel controller 101 and disk controller 103.

The disk controller 103 reads a data input/output request written in the shared memory 104 by the channel controller 101 and executes processing for writing, reading, and the like of data to and from the disk drives 108 according to a command specified by the data input/output request (for example, a SCSI (Small Computer System Interface) specification command). The disk controller 103 may also be provided with RAID (Redundant Arrays of Inexpensive Disks) functionality. In such a case, it distributes data among the disk drives 108 according to the RAID level (0, 1, or 5, for example).

The disk drives 108 may, for example, be hard disk devices. The disk drives 108 may be integrated into a disk array device or may exist separately. The data storage area provided by the disk drives 108 is administrated in units of physical volumes or logical volumes that are logically set up on

the physical volumes. Data can be read or written from or to the disk drives 108 by specifying an identifier of the logical volume and the location in it.

The management console 107 is a computer for maintaining/administering the disk array device or disk drives 108. For example, changing ~~of~~ the software ~~or parameters~~ executed in the channel controller 101 or disk controller 103 and/or its parameters, configuring the disk drives 108, administration or setting of logical volumes (capacity administration or capacity expansion/reduction, provisioning of them to information processing devices 10 and 20, and the like), and other operations are performed according to instructions from the management console 107. The management console 107 can be housed in the disk array device, or can exist separately.

The remote communication interface 102 is a communication interface (channel extender) for transferring data to the other storage device~~s~~ 30 and 40, and the transfer of replication data for the remote copying described hereinafter is performed via the remote communication interface 102. The remote communication interface 102 converts the interface of the channel controller 101 (for example, Fibre Channel, ESCON (registered trademark), FICON (registered trademark), or other interface) to the communication format of the first network 80. Data is thereby transferred to the other storage devices 30 and 40.

Also, the disk array device may function as a~~n~~ NAS (Network Attached Storage) device or the like configured to receive a data input/output request specifying a file name ~~assignment~~ from the information processing devices 10 and 20 by means of a~~n~~ NFS (Network File System) or other protocol, instead of the configuration described above, for example.

The shared memory 104 can be accessed from both the channel controller 101 and the disk controller 103. Other than being used for

exchanging data input/output request commands, the memory is also used to store administration information for the storage devices 30 and 40 or disk drives 108.

FIG. 3 is a block diagram depicting an example of the structure of the information processing devices 10 and 20 pertaining to the present embodiment.

The information processing devices 10 and 20 are provided with a CPU (processor) 110, memory 120, port 130, storage device 140, storage medium reading device 160, input device 170, output device 180, timer 200, and the like.

The CPU 110 administers control of the information processing devices 10 and 20 as a whole, performs various functions by executing programs stored in the memory 120, and performs ~~the~~ processing of the information processing devices 10 and 20 described hereinafter. These functions consist, for example, of operating an automatic teller system for a bank, operating a system for reserving airline seating, or the like.

The storage medium reading device 160 consists of a device for reading programs or data stored in the storage medium 190. The programs or data thus read are stored in the memory 120 and/or storage device 140. A flexible disk, CD-ROM, semiconductor memory, or the like may be used as the storage medium 190. The storage medium reading device 160 may be housed in the information processing devices 10 and 20, or may be external.

An operator or the like uses the input device 170 to input data and the like to the information processing devices 10 and 20. The input device 170 consists, for example, of a keyboard, mouse, or the like. The output device 180 is a device for outputting information to the outside. The output device 180 consists, for example, of a display, printer, or the like. The port 130 is a device for performing communication with the storage devices 30 and 40. This port may also be used for performing communication

between other information processing devices 10 and 20. In this case, the port 130 has a NIC (Network Interface Card) and an FC HBA (Fibre Channel Host Bus Adapter), for example. Consequently, the information processing devices 10 and 20 can be configured so as to receive programs or data stored in the memory 120 or storage device 140 of other information processing devices 10 and 20 via the port 130, and to store them in the memory 120 or storage device 140 of its own.

The timer 200 monitors the time for performing a number of functions in the present embodiment. The timer 200 consists, for example, of a hardware timer, software timer, or the like. By means of this timer 200, it becomes possible for the information processing devices 10 and 20 to determine that a timeout error has occurred when data processing or a data transfer has not been completed within a pre-set time period, and to abort the processing or communication and to execute recovery processing. Consequently, a system down condition occurring as a result of leaving a timeout unnoticed can be prevented. These structures are interconnected by means of a bus 150.

FIG. 4 is a block diagram depicting an example of the structure of the channel controller 101 pertaining to the present embodiment.

The channel controller 101 comprises a CPU 211, cache memory 212, control memory 213, ports 215, bus 216, and the like.

The CPU 211 administers control of the channel controller 101 as a whole, and performs the processing of the channel controller 101 described hereinafter by executing a program stored in the control memory 213. A control program (replication processing function) 214 stored in the control memory 213 executes remote copying described hereinafter by being executed by the CPU 211 of the storage devices 30 and 40. The cache memory 212 consists of memory for temporarily storing data, commands, or the like exchanged with the information processing devices 10 and 20. The port 215

consists of a communication interface for performing communication with the information processing devices 10 and 20. These structures are interconnected by means of the bus 216.

FIG. 5 is a diagram outlining the structure of the storage system described as an embodiment of the present invention.

Hardware, software, or the like for operating a cluster 310 is introduced into the first information processing device 10 and second information processing device 20 for the purpose of creating high availability. The cluster in the present embodiment mainly consists of a failover-type cluster. Configuring a failover-type cluster entails configuring two or more information processing devices to operate one at a time using one information processing device as the main (primary) and the other information processing device as an auxiliary (secondary), such that when a failure occurs in the main information processing device for whatever reason, the other information processing device takes over the processing that was being performed by the main information processing device. By means of this cluster 310 of the information processing devices 10 and 20, it becomes possible for the information processing devices 10 and 20 to monitor each other for failure in the other information processing devices 10 and 20 via the third network 70. The cluster 310 of the information processing devices 10 and 20 also allows the information processing devices 10 and 20 to take over (failover) the processing 300 that was being performed by other information processing devices 10 and 20 when a failure is detected in the other information processing devices 10 and 20. Also, the cluster 310 of the information processing devices 10 and 20 allows the first information processing device 10 to take over (failback) the processing 300 that was being performed by the second information processing device 20 when it is concluded that a failure of the first information processing device 10 has been recovered. Other than detecting

recovery and performing failback automatically as described above, failback can also be set to occur in cases in which the first information processing device 10 detects a failure in the second information processing device 20, in cases in which there is a failback request by the system administrator via the input device 170, and in other cases.

The information processing devices 10 and 20 are capable of transmitting a data read request or data write request to the storage devices 30 and 40 respectively ~~by means of the agent 320~~, and are capable of receiving a notice that the reading or writing of data from or to the storage devices 30 and 40 has been completed.

The first storage device 30 comprises a differential administration table 1 (400), first storage area 411, and the like. The second storage device comprises a second storage area 412 and the like.

The differential administration table 1 (400) consists of information for indicating that a replication of the data written in the first storage area 411 has not yet been transmitted to the second storage device 40, and therefore the replication of the data has not been written to the second storage area 412. Also, the differential administration table 1 (400) of the second storage device 40 is a copy of differential administration table 1 (400) in the first storage device 30 transmitted from the first storage device 30 during failover. The differential administration table 2 (401) consists of information for indicating that the data written in the second storage area 412 and third storage area 413 have not yet been transmitted to the first storage device 30, and therefore the data have not been written to the first storage area 411. The differential administration table 2 (401) may be created in advance, for example, during the boot process of the second storage device 40, or may be created during failover. Also, the differential administration table 2 (401) of the first storage device 30 is copied from that in the second storage device 40 during failback. By means of these

differential administration tables 1 and 2 (400 and 401), the storage devices 30 and 40 can determine the presence of untransferred data to the other storage devices 30 and 40 when performing the remote copying described hereinafter. By means of the differential administration tables 1 and 2 (400 and 401), the channel controller 101 of the storage devices 30 and 40 can also keep track of which blocks of data reside in which storage device.

FIG. 6 depicts an example of the differential administration tables 1 and 2 (400 and 401). A "1" or "0" is recorded in the bit value column of the differential administration tables 1 and 2 (400 and 401). When a "1" is recorded, this indicates that this block contains data that have not yet been transferred to the other storage devices 30 and 40. Alternatively, when a "0" is recorded, this indicates that the data in this block has been transferred to the other storage devices 30 and 40. Also, all the entries of each differential administration tables 1 or 2 (400 or 401) are initialized as '0' when it is created. In the present embodiment, the differential administration tables 1 and 2 (400 and 401) are stored in the shared memory 104, but may also be stored in storage areas 411 through 414, or in another location.

The first storage area 411 stores the data written by the first information processing device 10 or received from second storage device 40. The fourth storage area 414 consists of an auxiliary storage area. The second storage area 412 stores the data written by the second information processing device 20 or received from first storage device 30. The third storage area 413 consists of an auxiliary storage area. Also, in the present embodiment, storage areas 411 through 414 are all configured with the same capacity size, but are not limited to this arrangement.

The processing performed in the storage devices 30 and 40 when the storage devices 30 and 40 receive a data input/output request from the information processing devices 10 and 20 will next be described. When a

data write request transmitted from the information processing devices 10 and 20 is received, the channel controller 101 stores a command for a data write request (hereinafter referred to as "data write command") in the shared memory 104, and stores the target data of this data write command (hereinafter referred to as "write data") in the cache memory 105. The disk controller 103 monitors the contents of the shared memory 104 in real time. When this monitoring detects that a data write command has been written to the shared memory 104, the disk controller 103 reads the write data from the cache memory 105 and writes the write data to the storage areas 411 through 414 based on an address (block number) specified by the data write command.

When writing of data to the storage areas 411 through 414 is completed, the disk controller 103 notifies the channel controller 101 of this completion. When notice of the data write completion is received, the channel controller 101 transmits a notice of data write completion to the information processing devices 10 and 20.

When the channel controller 101 receives the aforementioned notice from the disk controller 103, the channel controller 101 updates the bit value column of the differential administration table 1 (400) to "1" based on the block number to which the data were written. An arrangement may also be adopted whereby the disk controller 103 updates the bit value column. The channel controller 101 then transmits a notice of data write completion to the information processing devices 10 and 20.

On the other hand, when a data read request transmitted from the information processing devices 10 and 20 is received, the channel controller 101 delivers a command for a data read request (hereinafter referred to as "data read command") to the disk controller 103. Also, the data read command can be transmitted from the channel controller 101 to the disk controller 103 via the shared memory 104.

When a data read command is received from the channel controller 101, the disk controller 103 reads the data to be read (hereinafter referred to as "read data") from the storage areas 411 through 414 based on an address specified by the data read command. The data thus read are then written to the cache memory 105. When transfer of data to the cache memory 105 is completed, the disk controller 103 notifies the channel controller 101 of this completion. The channel controller 101 then transmits the read data stored in the cache memory 105 to the information processing devices 10 and 20.

Also, besides data are exchanged between the first information processing device 10 and first storage device 30 in the storage system 90 pertaining to the present embodiment as described above, processing for managing the replication (remote copying) of the data in the first storage device 30 to the second storage device 40 is also performed in the background. By this remote copying, the data contentS in the storage areas 411 and 412 becomes identical, and thus redundant data management is achieved. Remote copying will be described hereinafter.

=== Remote Copying ===

FIG. 7 is a flowchart describing the processing relating to remote copying from the first storage device 30 (replication source storage device) to the second storage device 40 (replication destination storage device) pertaining to the present embodiment.

After transmitting data write completion notice to the first information processing device 10, the channel controller 101 of the first storage device 30 refers the differential administration table 1 (400) recorded in the shared memory 104 and transmits a write request for data not yet transferred to the second storage device 40 (S700). Also, the block number in the first storage area 411 in which the transmitted data are stored is set as the address specified by this data write command.

When the channel controller 101 of the second storage device 40 receives a data write command from the first storage device 30 (S701), the disk controller 103 writes the write data to the corresponding block in the second storage area 412 according to the data write command (S702). When writing of data to the second storage area 412 is completed, the channel controller 101 of the second storage device 40 transmits notice of data write completion to the first storage device 30 (S703).

When the notice of data write completion is received (S704), the channel controller 101 of the first storage device 30 updates, based on the block number specified by the data write request, the bit value in the differential administration table 1 (400) corresponding to that block from "1" to "0." Also, the updating may be performed by means of the disk controller 103.

The channel controller 101 of the first storage device 30 then refers the updated differential administration table 1 (400) and determines whether or not there are data that have not yet been transmitted to the second storage device 40 (S705). When the channel controller 101 of the first storage device 30 concludes that there are data that have not yet been transmitted to the second storage device 40 (S705; YES), the process proceeds to step (S700). On the other hand, when the channel controller 101 of the first storage device 30 concludes that there are no data that have not yet been transmitted to the second storage device 40 (S705; NO), the process is terminated.

As described above, by means of processing whereby the first storage device 30 presents the second storage device 40 with a replication of the data stored in the first storage area 411 and the second storage device 40 that received the replication of the data in the second storage area 412 stores the replication (hereinafter referred to as "first processing"), it becomes possible that the contents of the first storage area 411 and second

storage area 412 are the same, and that data redundancy is achieved.

Also, when the first processing is completed after failover is performed, processing is first performed whereby a replication of all the data which the corresponding bit of differential administration table 2 (401) is "1" and was stored by the disk controller 103 of the second storage device 40 in the third storage area 413 is written to the second storage area 412. Processing similar to steps (S700) through (S705) described above is then performed. The difference is that the direction of data flow is reversed. Namely, data is transferred from the second storage device 40 to the first storage device 30. Therefore, in (S700), the channel controller 101 of the second storage device 40 refers the differential administration table 2 (401) and transmits a write request for data not yet transmitted to the first storage device 30.

To determine the first processing is finished, for example, the first storage device 30 adds to the write request the last data mark which indicates there is no untransferred data remained in the first storage device 30. When the second storage system 40 receives the last data mark, it clears all the bits of the differential administration table 1 (400) to "0".

In a similar way, when the second processing is completed after failback is performed, a processing is performed whereby all the data whose corresponding bits of differential administration table 3 (402) are "1" and were stored in the fourth storage area 414 are copied to the first storage area 411. In this case, when each copy is performed, the corresponding bit of the differential administration table 3 (402) is reset to "0" and the corresponding bit of the differential administration table 1 (400) is set to "1". These bit operations are performed to make the copied data to the first storage area 411 the targets of remote copying. Make sure that this kind of bit operations to the differential administration table are not performed when the first processing is completed after failover is performed.

In this manner, by means of processing whereby the second storage device 40 presents the first storage device with the replication of the data stored in the second storage area 412 and the third storage area 413, and the first storage device 30 that received the replication stores the replication (hereinafter referred to as "second processing"), it becomes possible that the contents of the first storage area 411 and second storage area 412 are the same.

Also, when the second processing is completed after failback is performed, processing is first performed whereby a replication of all the data stored by the disk controller 103 of the first storage device 30 in the fourth storage area 414 is written to the first storage area 411. Processing in accordance with steps (S700) through (S705) is then performed. Also, in (S700), the channel controller 101 of the first storage device 30 refers the differential administration table 3 and transmits a write request for data not yet transmitted to the second storage device 40. The differential administration table 3 consists of information indicating that the data written in the first storage area 411 and the fourth storage area 414 have not yet been transmitted to the second storage device 40, and that the data are not written in the second storage area 412. The differential administration table 3 may be created in advance during the boot process of the first storage device 30, or may be created during failover after failback. The differential administration table 3 is transmitted from the first storage device 30 to the second storage device 40 during failover after failback. By means of this differential administration table 3, the second storage device 40 can determine whether or not data are present that are not yet transferred to the first storage device 30 when remote copying is performed. By means of the differential administration table 3, the channel controller 101 of second storage device 40 can also keep track of which blocks of data reside in which storage device. Also, the differential

administration table 3 has the same format as the differential administration tables 1 and 2 (400 and 401).

In this manner, by means of processing whereby the first storage device 30 presents the second storage device 40 with the replication of the data stored in the fourth storage area 414, and the second storage device 40 that received the replication stores the replication in the second storage area 412 (hereinafter referred to as "third processing"), it becomes possible that the contents of the first storage area 411 and second storage area 412 are the same.

=== Processing performed when failure occurs in the first information processing device ===

An example of the processing performed when the second information processing device 20 detects failure of the first information processing device 10 will next be described using FIG. 8.

When the second information processing device 20 detects failure of the first information processing device 10 (S800), the second information processing device 20 notifies the second storage device 40 that failover will be performed (S801).

When notice that failover will be performed is received from the second storage device 20 (S802), the channel controller 101 of the second storage device 40 requests the differential administration table 1 (400) from the first storage device 30 (S803).

The channel controller 101 of the first storage device 30 presents the second storage device 40 with a replication of the differential administration table 1 (400) stored in the shared memory 104 (S805) upon receipt of a request for the differential administration table 1 (400) from the second storage device 40 (S804).

The channel controller 101 of the second storage device 40 stores the differential administration table 1 (400) in the shared memory 104

(S807) upon receipt of the differential administration table 1 (400) from the first storage device 30 (S806). The channel controller 101 of the second storage device 40 then notifies the second information processing device 20 that a data input/output request can be received (S808).

The second information processing device 20 takes over the processing being performed by the first information processing device 10 (performs failover) (S810) upon receipt of notification from the second storage device 40 that a data input/output request can be received (S809), and completes the processing. The second information processing device 20 then presents the second storage device 40 with a data input/output request **issued from the failed-over processing**, and the second storage device 40 processes the data according to the data input/output request transmitted from the second information processing device 20.

Also, in the present embodiment, processing was described for a case in which the second information processing device 20 detects failure of the first information processing device 10, but processing **similar to** steps (S801) through (S810) is also performed in the case of failback. Also, notification that failback will be performed is made, for example, after failure of the first information processing device 10 has been recovered.

In the present embodiment, an arrangement is described in which the second storage device 40 receives the differential administration table 1 (400) from the first storage device 30 prior to failover, but the second storage device 40 may also receive the differential administration table 1 (400) from the first storage device 30 after failover.

In the present embodiment, the processing in steps (S803) through (S810) is performed when notice of failover is received, but the processing in steps (S803) through (S810) may also be performed in a case in which the first data input/output request is received from the information processing devices 10 and 20. **In this case, notice of failover is not issued, but the**

second storage device 40 can determine that failover has been made by receiving an input/output request from the second information processing device 20.

The present embodiment is configured such that the second information processing device 20 monitors failure of the first information processing device 10, but the first storage device 30 may also monitor failure of the first information processing device 10. In this case, an arrangement may be adopted whereby the first storage device 30 transmits the differential administration table 1 (400) to the second storage device 40 when the first storage device 30 detects failure of the first information processing device 10, and the second storage device 40 can transmit notice to the second information processing device 20 that a data input/output request can be received. By this means, the second information processing device 20 executes failover, and is able to transmit a data input/output request to the second storage device 40.

In the present embodiment, the storage devices 30 and 40 receive the differential administration tables 1 and 2 (400 and 401) by requesting them from the other storage devices 30 and 40, but the differential administration tables 1 and 2 (400 and 401) may also be presented to the other storage devices 30 and 40 each time they are updated by the storage devices 30 and 40. Also, an arrangement may be adopted whereby the differential administration tables 1 and 2 (400 and 401) transmitted to the other storage devices 30 and 40 from the storage devices 30 and 40 are presented via the information processing devices 10 and 20. The storage devices 30 and 40 may also transmit information to the other storage devices 30 and 40 the address (block number) information specified in data write requests issued by the information processing devices 10 and 20, instead of by the aforementioned differential administration tables 1 and 2 (400 and 401).

By means of the mechanism described above, timeouts that occur due

to the information processing devices 10 and 20 waiting for processing performed during failover or failback (processing for making the contents of the replication destination storage areas 411 (and 412) and the contents of the replication source storage areas 412 (and 411) are the same) can be avoided, and takeover of processing between the information processing devices 10 and 20 can be performed smoothly. It also becomes possible to smoothly operate the storage system, and to enhance the reliability and availability of the storage system.

=== Processing performed when a data write command is received ===

An example of the processing performed when the second storage device 40 receives a data write request from the second information processing device 20 after failover will next be described using FIG. 9.

When a data write request is received from the second information processing device 20 (S900), the channel controller 101 of the second storage device 40 refers the differential administration table 1 (400) and determines whether or not the bit values in the differential administration table 1 (400) are all "0" (S901). When the channel controller 101 of the second storage device 40 concludes that the bit values in the differential administration table 1 (400) are not all "0" (S901; NO), the process proceeds to step (S904). Alternatively, when the channel controller 101 of the second storage device 40 concludes that the bit values in the differential administration table 1 (400) are all "0" (S901; YES), the process proceeds to step (S902).

In step (S902), the channel controller 101 of the second storage device 40 determines whether or not the data stored in the third storage area 413 are stored in the second storage area 412 (S903). When the channel controller 101 of the second storage device 40 concludes that a replication of all the data stored in the third storage area 413 is not stored in the second storage area 412 (S902; NO), the process proceeds to step (S904).

Alternatively, when the channel controller 101 of the second storage device 40 concludes that a replication of all the data stored in the third storage area 413 is stored in the second storage area 412 (S902; YES), the channel controller 101 of the second storage device 40 stores a data write command in the shared memory 104 and stores this write data in the cache memory 105. The disk controller 103 reads the write data from the cache memory 105 upon detecting that a data write command has been written to the shared memory 104 and writes the write data to the second storage area 412 based on the block number specified by the data write command (S903).

In step (S904), the disk controller 103 of the second storage device 40 writes the write data **both** to the third storage area 413 **and to the second storage area 412** in the same manner as in step (S903).

When writing of data to the storage areas 412 **or** 413 is completed (S903 and S904), the disk controller 103 of the second storage device 40 notifies the channel controller 101 of the second storage device 40 of this completion. When notice of data write completion is received, the channel controller 101 of the second storage device 40 updates the block number bit value column of the differential administration table 2 to "1" based on the block number specified by the data write request (S905). The channel controller 101 of the second storage device 40 then notifies the second information processing device 20 that writing of data is completed (S906), and processing is terminated.

Also, in the present embodiment, the target data of the data write command are stored in the third storage area 413 in step (S904), but an arrangement may be adopted whereby the channel controller 101 of the second storage device 40 transfers a data write request to the first storage device 30, and the channel controller 101 of the first storage device 30 stores the data in the first storage area 411 according to the aforementioned data write request. In this case, after data writing is completed, the

channel controller 101 of the first storage device 30 updates the bit value column of the differential administration table 1 (400) to "1" based on the block number specified by the data write request. When notice is received from the first storage device 30 that data writing is completed, the channel controller 101 of the second storage device 40 also updates the bit value column of the differential administration table 1 (400) to "1" based on the block number specified by the data write request.

Also, as described above, when the first processing is completed, the disk controller 103 of the second storage device 40 stores a replication of the data stored in the third storage area 413 in the second storage area 412, and then initiates the second processing. By performing processing according to this sequence, the newest data can be secured in the second storage area 412, and the newest data can also be redundantly administrated in the first storage area 411 as well.

In the present embodiment, processing after failover was described, but processing is also performed in accordance with steps (S900) through (S906) after failback. In this case, the "third storage area 413" of step (S902) becomes the "fourth storage area 414," and the "differential administration table 2 (401)" of step (S905) becomes the "differential administration table 3." The differential administration table 3 consists of information indicating that a replication of the data written in the fourth storage area 414 has not yet been transmitted to the second storage device 40, and that a replication of the data has not been written to the second storage area 412. The differential administration table 3 is newly created during failback and is stored in the shared memory 104 by the channel controller 101 of the first storage device 30.

Processing is also performed in accordance with steps (S900) through (S906) in a case in which failback is performed during the first processing. In the background, as described above, after the first processing is completed,

the disk controller 103 of the second storage device 40 stores in the second storage area 412 a replication of the data stored in the third storage area 413. Then the second processing is initiated. And when it is completed, the disk controller 103 of the first storage device 30 stores in the first storage area 411 a replication of the data stored in the fourth storage area 414. The third processing is initiated thereafter. By processing according to this sequence, the newest data can be secured, and the newest data can also be redundantly administrated.

=== Processing performed when a data read request is received ===

An example of processing performed when the second storage device 40 receives a data read request from the second information processing device 20 after failover will next be described using FIG. 10.

When a data read request is received from the second information processing device 20 (S1000), the channel controller 101 of the second storage device 40 determines whether or not the data specified by the data read command are present in the third storage area 413 (S1001). This determination can be performed, for example, by referring the differential administration table 1 (400) and the differential administration table 2 (401). When the channel controller 101 of the second storage device 40 concludes that the data specified by the data read command are present in the third storage area 413 (S1001; YES), the process proceeds to step (S1008). Alternatively, when the channel controller 101 of the second storage device 40 concludes that the data specified by the data write command are not present in the third storage area 413 (S1001; NO), the channel controller 101 of the second storage device 40 determines whether or not the data specified by the data read command are present in the second storage area 412 (S1002). This determination may be performed, for example, by referring the differential administration table 1 (400). Also, performing a determination in accordance with the sequence described above enables the

second storage device 40 to determine where the newest data are.

In step (S1002), when the channel controller 101 of the second storage device 40 concludes that the data specified by the data read command are present in the second storage area 412 (S1002; YES), the process proceeds to step (S1009). Alternatively, when the channel controller 101 of the second storage device 40 concludes that the data specified by the data read command are not present in the second storage area 412 (S1002; NO), the channel controller 101 of the second storage device 40 presents the first storage device 30 with a data read request (S1003).

When a data read request is received from the second storage device 40 (S1004), the channel controller 101 of the first storage device 30 delivers a data read command to the disk controller 103. Upon receipt of the data read command from the channel controller 101, the disk controller 103 reads the read data from the first storage area 411 based on the block number specified by the data read command (S1005). This read data is then written to the cache memory 105. When data transfer to the cache memory 105 is completed, the disk controller 103 notifies the channel controller 101 of this completion. When the aforementioned notice is received from the disk controller 103, the channel controller 101 of the first storage device 30 transmits to the second storage device 40 the read data stored in the cache memory 105 (S1006).

When the read data is received from the first storage device 30 (S1007), the channel controller 101 of the second storage device 40 transmits the read data to the second information processing device 20 (S1010), and the process is terminated.

In step (S1008), when the disk controller 103 receives the data read command from the channel controller 101 of the second storage device 40, the disk controller 103 reads the read data from the third storage area 413 based on the block number specified by the data read command. The channel

controller 101 of the second storage device 40 then transmits the read data to the second information processing device 20 (S1010), and the process is terminated.

In step (S1009), the disk controller 103 reads the read data form the second storage area 412 in the same manner as in step (S1008). The channel controller 101 of the second storage device 40 then transmits the read data to the second information processing device 20 (S1010), and the process is terminated.

In the present embodiment, processing after failover was described, but processing is also performed in accordance with steps (S1000) through (S1010) after failback. In this case, in step (S1001), the channel controller 101 of the first storage device 30 determines whether or not the data specified by the data read command are present in the fourth storage area 414. This determination may be performed, for example, by referring **the differential administration table 1 and** the differential administration table 3. Also, in step (S1002), the channel controller 101 of the first storage device 30 determines whether or not the data specified by the data read command are present in the **first** storage area 411. This determination may be performed, for example, by referring the differential administration table 1.

Also, when it is concluded that the data specified by the data read command are **not** present in the **first** storage area 411 (S1002; **NO**), the channel controller 101 of the first storage device 30 presents the second storage device 40 with a data read request (S1003). Alternatively, when it is concluded that the data specified by the data read command are ~~not~~ present in the third storage area 413 (S1002; **YES**), the channel controller 101 of the first storage device 30 reads the data from the first storage area 411 (S1009). Also, **similar** processing is performed in a case in which failback is performed during execution of the first processing.

Performing a determination in accordance with the sequence described above enables the storage devices 30 and 40 to determine where the newest data are. Also, processing in the manner described above enables the storage devices 30 and 40 to provide the newest data in response to a data read request transmitted from the information processing devices 10 and 20.

=== Other embodiments ===

FIG. 11 is a diagram outlining the structure of the storage system 90 described as an example of another embodiment.

The storage devices 30 and 40 are provided with data location determination tables 500 and 600, local unreflected tables 501 and 601, remote unreflected tables 502 and 602, normal volumes 510 and 610, local unreflected volumes 511 and 611, remote unreflected volumes 512 and 612, and the like.

The normal volumes 510 and 610 consist of storage areas for writing the write data transmitted from the storage devices 30 and 40 or information processing devices 10 and 20. The local unreflected volumes 511 and 611 consist of auxiliary storage areas. When write data transmitted from the information processing devices 10 and 20 are written to the local unreflected volumes 511 and 611, the channel controllers 101 or disk controllers 103 of the storage devices 30 and 40 **search a free entry in** the local unreflected tables 501 and 601 stored in the shared memory 104 or the like, **create a new entry, and write the write data to the block corresponding to the newly created entry.**

An example of the local unreflected tables 501 and 601 is depicted in FIG. 12. A normal volume block number column, time column, and backward pointer column are established for each **entry** in the local unreflected tables 501 and 601. **And the entry number is a corresponding block number of the local unreflected volume 511 or 611.** The block number specified by the data write command transmitted from the information processing

devices 10 and 20 is recorded in the normal volume block number column. Also, in an initial state, a "-1" indicative of an empty entry is recorded in the normal volume block number column. By providing this normal volume block number column, the channel controller 101 or disk controller 103 of the storage devices 30 and 40 can keep track of the block numbers of the normal volumes 510 and 610 under which the data stored in the block numbers of the local unreflected volumes 511 and 611 must be stored.

In the time column, the time (for example, year, month, day, hour, minute, second **and millisecond**) recorded in the header of the data write command transmitted from the information processing devices 10 and 20 is recorded. Also, this time may consist, for example, of the time at which the information processing devices 10 and 20 created the data write request, or may consist of the time at which the information processing devices 10 and 20 transmitted the data write request to the storage devices 30 and 40. This time is monitored by the timer 200 of the information processing devices 10 and 20, and is recorded in the header of the data write command when this command is transmitted. Also, the channel controller 101 of the storage devices 30 and 40 may record in the time column the time at which the data write command was received from the information processing devices 10 and 20. Providing this time column enables the channel controller 101 of the storage devices 30 and 40 to administrate the write data received from the information processing devices 10 and 20 in chronological order.

The rear pointer column has the next entry number. In this way, valid entries of local unreflected table 501, 601 comprise a queue. In the queue, the entries are sorted based on "time" column values and the newest entry comes last. Putting a newly created entry in the queue is performed by comparing the "time" column of the new entry and the "time" columns of entries in the queue. The newly created entry is inserted in the position where

all the entries including the newly created one are sorted in the order of "time."

If a newer entry than the newly created one is not found, a value "-1" is set to the rear pointer column of the newly created entry. Providing this rear pointer enables the disk controller 103 of the storage devices 30 and 40 to store the data stored in the local unreflected volumes 511 and 611 in order from oldest to newest in the normal volumes 510 and 610 before remote copying is executed, and to preserve consistency of data.

The remote unreflected volumes 512 and 612 consist of storage areas that contain data to be transmitted to the other storage devices 30 and 40 in order to make the contents of the normal volumes 510 and 610 of the storage devices 30 and 40 are the same.

When the storage devices 30 and 40 write to the normal volumes 510 and 610 or local unreflected volumes 511 and 611 the write data transmitted from the information processing devices 10 and 20, the storage devices 30 and 40 write the write data also to the remote unreflected volumes 512 and 612. The channel controller 101 of the storage devices 30 and 40 search a free entry in the remote unreflected tables 502 and 602 stored in the shared memory 104 or the like, create a new entry, and write the write data to the block corresponding to the newly created entry. This updating may also be performed by means of the disk controller 103 of the storage devices 30 and 40.

FIG. 13 depicts an example of the remote unreflected tables 502 and 602. A normal volume block number column, time column, and backward pointer column are established for each entry of the remote unreflected tables 502 and 602, and the same contents are recorded in those columns as in the case of the local unreflected tables 501 and 601. And the entry number is a corresponding block number of the remote unreflected volume 512 or 612. The channel controller 101 of the storage devices 30 and 40 can transmit

to the other storage devices 30 and 40 the data stored in the remote unreflected volumes 512 and 612 in order from oldest to newest during remote copying by referring the remote unreflected tables 502 and 602. Also, because the storage devices 30 and 40 are able to store the write data thus transmitted from oldest to newest in the normal volumes 510 and 610, the consistency of the data can be preserved.

By referring to the data location determination tables 500 and 600, it is possible to determine where the newest data is. Prototypes of the data location determination tables 500 and 600 are created by the channel controller 101 of the storage devices 30 and 40 when it receives a transmission request from the other storage devices 30 and 40 ~~is present~~, and then it transmits the table to the other storage devices 30 and 40. The prototypes of the data location determination tables 500 and 600 are created based on the remote unreflected tables 502 and 602, and by reflecting the status of the local unreflected table 501, 601 to the prototype by the channel controller 101 in the storage device 30, 40 which received the prototype, the prototype becomes a full-fledged data location determination table 500 or 600 which can be used to determine where newest data can be found. Also, a request for transmission of the prototype of the data location determination tables 500 or 600 is issued, for example, when there is notice of failover or failback from the information processing devices 10 or 20 that access the storage devices 30 or 40.

FIG. 14 depicts an example of the data location determination tables 500 and 600. The way the data location determination table 500, 600 is created will be described here. First of all, the channel controller 101 in the storage device 30, 40 which received a request to send the prototype of the data location determination table 500, 600 initializes all the columns of the prototype to "-1". Then for each entry of the remote unreflected table 502, 602, if a block number is recorded in the normal volume block

number column, the channel controller 101 of the storage devices 30 or 40 records in the prototype's column that corresponds to that block number the value recorded in the "time" column of the entry of the remote unreflected table 502, 602.

After this, the channel controller 101 sends the prototype to the other storage device 30, 40 which issued the request of a prototype.

The channel controller 101 in the storage device 30, 40 which received the prototype executes the following procedures for each entry of the prototype:

(1) If the value stored in the entry is "-1", the controller looks up the local unreflected table 501, 601 to find "corresponding" entries. "Corresponding" here means that the entry number of the prototype and the value stored in the normal block number column of the local unreflected table 501, 601 are the same. If there is one or more corresponding entries, choose the newest entry among the corresponding entries and set the entry number to the prototype's entry.

(2) If the value stored in the entry is not "-1", the controller looks up the local unreflected table 501, 601 to find "corresponding" entries. "Corresponding" here has the same meaning as in (1). If there isn't a corresponding entry, the controller sets "-2" to the entry of the prototype. If there is one or more corresponding entries, the controller chooses the newest entry among the corresponding entries and compares the value stored in the "time" column of the newest entry and the value stored in the prototype's entry. If the value stored in the prototype is

newer, the controller sets "-2" to the prototype's entry. If the value stored in the prototype isn't newer, the controller sets to the prototype's entry the aforementioned newest entry number.

In this way, the prototype becomes the full-fledged data location determination table 500 600.

Thus based on the data location determination tables 500 and 600 created in this manner, the channel controller 101 of the storage devices 30 and 40 that received the prototype can keep track of which volume contains the newest target data of a data read request transmitted from the information processing devices 10 and 20.

Specifically, if "-1" is stored in the entry, the newest data is in the normal volume 510, 610. If "-2" is stored, the newest data is in the remote storage device 30, 40. If the other values are stored, the newest data is in the local unreflected volume 511, 611 and the value indicates the block number in it.

Also, in the present embodiment, when data corresponding to a data write request transmitted from the information processing devices 10 and 20 are written to the local unreflected volumes 511 and 611, the storage devices 30 and 40 that received the data location determination tables 500 and 600 record the local unreflected volume 511's or 611's block number in which the data are written in the data location determination table 500's or 600's column that corresponds to the normal volume block number specified by the data write command. By this means, the channel controller 101 of the storage devices 30 and 40 can keep track of the volumes that contain the target data of the data read command transmitted from the information processing devices 10 and 20. Also, when the data location determination tables 500 or 600 is received, the storage devices 30 and 40 may be configured so as to store the tables in the shared memory 104, or may be configured

so as to store the tables in other storage area (memory, volume, or the like).

=== Remote copying ===

FIG. 15 is a flowchart depicting the remote copying performed in the background, described as an example of another embodiment.

After notification of data write completion is transmitted to the first information processing device 10, the channel controller 101 of the first storage device 30 refers the valid entries' time column of the remote unreflected table 502 stored in the shared memory 104, creates a write request for the oldest data stored in the remote unreflected volume 512, and transmits the request to the second storage device 40 (S1500). Also, the block number recorded in the normal volume block number column of the remote unreflected table 502 is recorded as the target address of the data write command. **At the same time, the value in the "time" column of the remote unreflected table 502 of the oldest data, a flag which indicates if there is at least one untransferred data to the remote storage device 40 ("the total remaining data flag" hereafter) and an information which indicates if there is one or more untransferred data whose target address is the same as the one recorded in the data write command ("the remaining data information" hereafter) are transferred as a part of the write request. The total remaining data flag is either "1" (Yes) or "0" (No). The remaining data information is "0" (No data) if there is no remaining data. Or the information is the "time" value of the newest untransferred data aimed at the target address if there is one or more remaining data. Both the total remaining data flag and the remaining data information can be created from the remote untransferred table 502.**

The processing in steps (S1501) through (S1504) is then performed. Also, because the processing of the step (S1501) is performed in the same manner as in the step (S701), description thereof is omitted.

Then the step (S1502) will be described. The second storage device 40 writes the received data to free space in the local unreflected volume 611. The local unreflected table 601 is searched for an entry whose normal block number column value is "-1" to find the free space. Of course, the queue made up of valid entries of the local unreflected table 601 is looked up from the oldest entry and the newly created entry is inserted in the correct position so that all the entries in the queue are in the order of time. Then for every entry from the oldest to the newly created one in the queue, corresponding data in the local unreflected volume 611 is copied to the normal volume 610. When each copy is finished, the normal block number column of the entry in the local unreflected table 601 is reset to "-1". This means the entry is invalidated. If the total data remaining flag is "0" (No), additional data reflection from the local unreflected volume 611 to the normal volume 610 is made for all the entries in the queue from the oldest to the newest. Do not forget the entry invalidation after each copy.

For the remote copy data that was transferred from the first storage device 30, look up the queue of the local unreflected table 601 to find the newest entry and update the corresponding entry of the data location determination table 600 with the following procedure:

(1) If there is no unreflected data and the remaining data information is "0" (No data), set "-1" to the corresponding entry of data location determination table 600.

(2) If there is no unreflected data and the remaining data information is NOT "0", set "-2".

(3) If there is one or more unreflected data and the remaining data information is "0", set the entry number of the newest entry.

(4) If there is one or more unreflected data and the remaining data information is NOT "0", compare the "time" column of the newest entry and the remaining data information.

(4-1) If the remaining data information is newer, set "-2".

(4-2) If the remaining data information is older, set the number of the newest entry.

For each reflected data from the local unreflected volume 611 to the normal volume 610 after writing the remote copy data to the local unreflected volume 611, update the corresponding entry of the data location determination table 600 by the following procedure:

(1) If the corresponding entry of the location determination table 600 is "-2", leave it as it is.

(2) Look up the local unreflected table 601 to find the corresponding entries and choose the newest one.

(3) If no corresponding entry is found, set "-1".

(4) If one or more corresponding entries are found, set the number of the newest entry.

If the storage device 40 does not have the data location determination table 600, the updates of the table described above are not performed.

As the processing of the step (S1503) through (S1504) is performed in the same manner as in the step (S703) through (S704), description thereof is

omitted.

The channel controller 101 of the first storage device 30 then verifies the rear pointer column of the remote unreflected table 502 and determines whether or not the rear pointer is "-1" (S1505). When it is concluded that the rear pointer is not "-1" (S1505; NO), the process proceeds to step (S1500), and, based on the block number recorded in the rear pointer column of the remote unreflected volume 512, a write request for the data stored in that block (oldest data) is transmitted to the second storage device 40 (S1500). Alternatively, when it is concluded that the rear pointer is "-1" (S1505; YES), the channel controller 101 of the first storage device 30 concludes that there are no data not yet transferred to the second storage device 40, and the process is terminated.

After the determination in step (S1505) is performed, the channel controller 101 of the first storage device 30 updates the normal volume block number column of the target entry of the remote unreflected table 502 to "-1"

As described above, by means of processing whereby the first storage device 30 transmits to the second storage device 40 the data stored in the remote unreflected volume 512, and the second storage device 40 that received these data stores the aforementioned data in the normal volume 610 or in the local unreflected volume 611, and also data is reflected from the local unreflected volume 611 to the normal volume 610, the contents of the normal volumes 510 and 610 can be the same eventually (when remote copying between the storage devices 30, 40 are done and there is no remote copy pending data remained in either storage device 30, 40).

=== Processing performed when failure occurs in the first information processing device ===

When the second information processing device 20 detects a failure

of the first information processing device 10, processing is performed in accordance with the flow of steps (S800) through (S810) shown in FIG. 8.

Also, in step (S803), the channel controller 101 of the second storage device 40 requests the prototype of the data location determination table 500 from the first storage device 30.

In step (S805), the first storage device 30 creates a prototype of data location determination table 500 on the basis of the remote unreflected table 502 as described above, and transmits it to the second storage device 40. Then the channel controller 101 in the second storage device 40 creates the data location determination table 600 based on the prototype.

Also, processing was described in the present embodiment for a case in which the second information processing device 20 detects a failure of the first information processing device 10, but processing is also performed in accordance with steps (S801) through (S810) in the case of failback. In the present embodiment, failover and failback can be performed repeatedly.

=== Processing performed when a data write request is received ===

An example of processing performed in a case in which the storage devices 30 and 40 receive a data write request from the information processing devices 10 and 20 after failover or failback will next be described using FIG. 16.

When a data write request is received from the information processing devices 10 and 20 (S1600), the channel controller 101 of the storage devices 30 and 40 refers the data location determination tables 500 and 600 and determines whether the value stored in the corresponding entry is other than "-1" or not (S1601). When it is concluded that the value is other than "-1" (S1601; YES), the process proceeds to step (S1603). Alternatively, when it is concluded that the value is "-1" (S1601; NO), the process proceeds to step (S1602).

In step (S1602), the channel controller 101 of the storage devices 30

and 40 stores a data write command in the shared memory 104 and stores the write data in the cache memory 105. Upon detecting that a data write command has been written to the shared memory 104, the disk controller 103 reads the write data from the cache memory 105 and writes the write data to the normal volumes 510 and 610 based on the block number specified by the data write command.

In step (S1603), as in step (S1602), the disk controller 103 of the storage devices 30 and 40 writes the write data to an open area of the local unreflected volumes 511 and 611. When the channel controller 101 receives notice of data write completion from the disk controller 103 thereafter, the channel controller 101 updates the contents of the local unreflected tables 501 and 601 (S1604). This updating is performed **to the entry whose entry number is the same as** the number of the block **of the local unreflected volume 511, 611** to which the data was written. The address specified by the data write command is recorded in the normal volume block number column of the local unreflected tables 501 and 601. The time recorded in the header of the data write command is also recorded in the time column of the local unreflected tables 501 and 601. The channel controller 101 then updates the backward pointer column while referring the time column of the local unreflected tables 501 and 601, so that the data stored in the local unreflected volumes 511 and 611 are stored in order from oldest to newest in the normal volumes 510 and 610.

Then the channel controller 101 writes the entry number of the newly created entry of the local unreflected table 511, 611 in the corresponding column of the data location determination table 500, 600 to specify where the newest data is located.

In step (S1605), the disk controller 103 of the storage devices 30 and 40 writes to the remote unreflected volumes 512 and 612 the data written in the normal volumes 510 and 610 or local unreflected volumes 511 and 611.

When the channel controller 101 receives notice of data write completion from the disk controller 103 in the same manner as in step (S1604), the channel controller 101 then updates the contents of the remote unreflected tables 502 and 602 (S1606). The channel controller 101 of the storage devices 30 and 40 then notifies the information processing devices 10 and 20 that writing of data is completed (S1607), and the process is terminated.

Processing performed when a data read request is received

After failover or failback, the same processing as in steps (S1000) through (S1010) shown in FIG. 10 is performed in a case in which the channel controller 101 of the storage devices 30 and 40 receives a data read request from the information processing devices 10 and 20.

Also, in step (S1001), the channel controller 101 of the storage devices 30 and 40 determines whether or not the newest data specified by the data read command received from the information processing devices 10 and 20 are present in the local unreflected volumes 511 and 611. In step (S1002), the channel controller 101 of the storage devices 30 and 40 also determines whether or not the newest data specified by the data read command are present in the normal volumes 510 and 610. These determinations are performed by referring the value of columns that correspond to the normal volume block number of the data location determination tables 500 and 600, based on the address specified by the data read command.

When the block number of the local unreflected volumes 511 and 611 is recorded in the aforementioned column, the channel controller 101 of the storage devices 30 and 40 concludes that the newest data are present in the local unreflected volumes 511 and 611 (S1001; YES). When "-1" is recorded in the aforementioned column, the channel controller 101 of the storage devices 30 and 40 also concludes that the newest data are present in the normal volumes 510 and 610 (S1002; YES). Also, because the channel controller 101 of the storage devices 30 and 40 concludes that the newest

data are present in the remote unreflected volumes 512 and 612 of the other storage devices 30 and 40 when "-2" is recorded in the aforementioned column, in step (S1005), the other storage devices 30 and 40 read from the remote unreflected volumes 512 and 612 the newest target data of the data read request.

The present embodiment was described above, but the above examples were given in order to aid in understanding the present invention, and are not to be interpreted as limiting the present invention. The present invention may be modified or improved without deviating from the essence thereof, and its equivalents are also encompassed by the present invention.